Why Spring Boot?

- Spring based applications have a lot of configuration.
- When we use Spring MVC, we need to configure component scan, dispatcher servlet, a view resolver, web jars(for delivering static content) among other things.

Why Spring Boot?

- World is moving towards Microservices.
- We do not have a lot of time to set up 100 microservices.

Spring Boot Goals

- Quick Start to Spring
- Be opinionated
- Non functional features
- No code generation

Spring Boot Features

- Auto Configuration
- Spring Boot Starter Projects
- Spring Boot Actuator
- Embedded Server

Questions

- Compare Spring Boot vs Spring?
- Compare Spring Boot vs Spring MVC?

Spring

Most important feature of Spring Framework is Dependency Injection. At the core of all Spring Modules is Dependency Injection or IOC Inversion of Control.

```java
@Component
public class WelcomeService {
    //Bla Bla Bla
}

@RestController
public class WelcomeController {

    @Autowired
    private WelcomeService service;

  @RequestMapping("/welcome")
  public String welcome() {
    return service.retrieveWelcomeMessage();
  }
```

Problems Spring Solves

- Problem 1 : Duplication/Plumbing Code
- Problem 2 : Good Integration with Other Frameworks.

Spring MVC

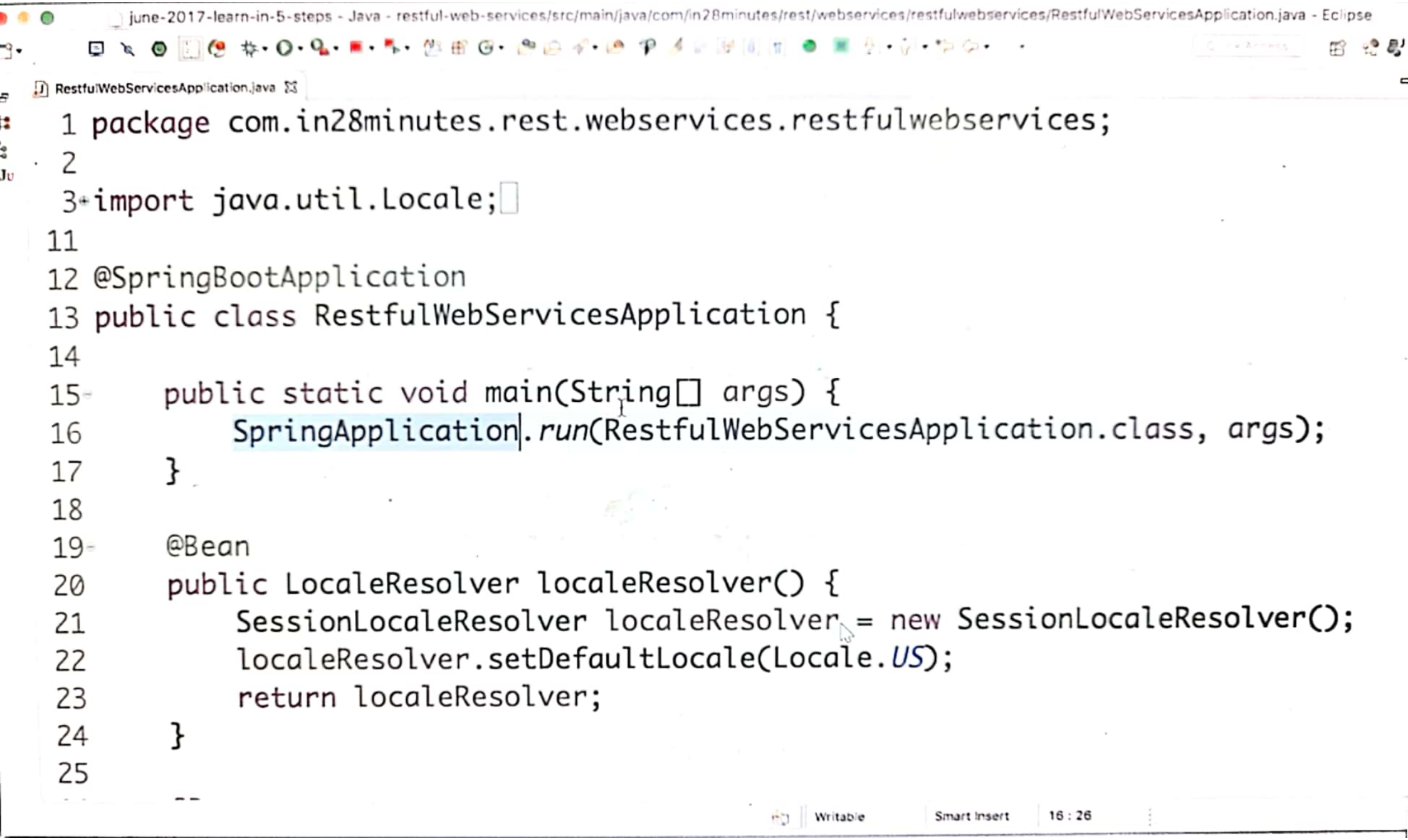
Spring MVC Framework provides decoupled way of developing web applications. With simple concepts like Dispatcher Servlet, ModelAndView and View Resolver, it makes it easy to develop web applications.

Spring Boot

- Eliminates all configuration needed by Spring and Spring MVC and auto configures it
  - No need for @ComponentScan. Default Component Scan.
  - No need to configure DispatcherServlet
  - No need to configure a Data Source, Entity Manager Factory or Transaction Manager.

Spring Boot Thinking

Can we bring more intelligence into this? When a spring mvc jar is added into an application, can we auto configure some beans automatically?



SpringBootApplication

```
@SpringBootConfiguration
@EnableAutoConfiguration
@ComponentScan
public @interface SpringBootApplication {
```

Questions

- What is Auto Configuration?
- How can we find more information about Auto Configuration?

Spring Boot looks at a) Frameworks available on the CLASSPATH b) Existing configuration for the application. Based on these, Spring Boot provides basic configuration needed to configure the application with these frameworks. This is called Auto Configuration.

Application Startup Log

```
Mapping servlet: 'dispatcherServlet' to [/]

Mapped "{[/error]}" onto public org.springframework.http.Respo
<java.util.Map<java.lang.String, java.lang.Object>> org.spring
BasicErrorController.error(javax.servlet.http.HttpServletReque

Mapped URL path [/webjars/* *] onto handler of type
[class org.springframework.web.servlet.resource.ResourceHttpRe
```

Questions

- What is an embedded server? Why is it important?
- What is the default embedded server with Spring Boot?
- What are the other embedded servers supported by Spring Boot?

Embedded Server

- Server is embedded as part of the deployable - jar.
- Removes the need to have the server pre-installed on the deployment environment.
- Default is Tomcat.
- Spring Boot also supports Jetty and UnderTow.

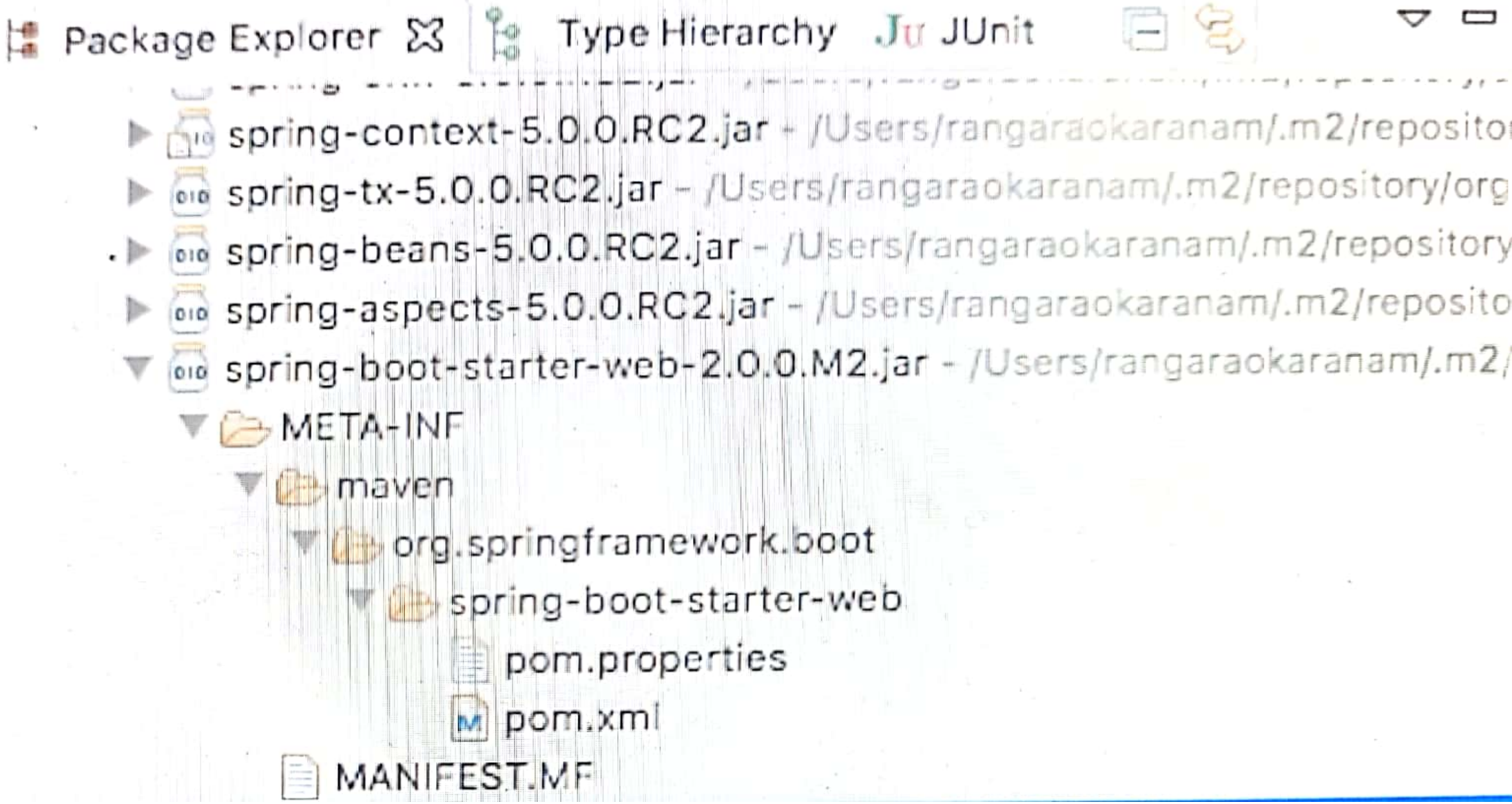
Switching to Jetty

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    <exclusions>
        <exclusion>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-tomcat</artifactId
        </exclusion>
    </exclusions>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jetty</artifactId>
</dependency>
```

Questions

- What are Starter Projects?
- Can you give examples of important starter projects?

36. 306-Introduction to Starter Projects



```
ependencies>
 <dependency>
     <groupId>org.springframework.boot</groupId>
     <artifactId>spring-boot-starter</artifactId>
 </dependency>
 <dependency>
     <groupId>org.springframework.boot</groupId>
     <artifactId>spring-boot-starter-json</artifactId>
 </dependency>
 <dependency>
     <groupId>org.springframework.boot</groupId>
     <artifactId>spring-boot-starter-tomcat</artifactId>
 </dependency>
 <dependency>
     <groupId>org.hibernate</groupId>
     <artifactId>hibernate-validator</artifactId>
 </dependency>
 <dependency>
```

Spring Boot Documentation

Starters are a set of convenient dependency descriptors that you can include in your application. You get a one-stop-shop for all the Spring and related technology that you need, without having to hunt through sample code and copy paste loads of dependency descriptors.

Starters

- spring-boot-starter-web-services - SOAP WebServices
- spring-boot-starter-web - Web & RESTful applications
- spring-boot-starter-test - Unit, Integration Testin
- spring-boot-starter-jdbc - Traditional JDBC
- spring-boot-starter-hateoas - HATEOAS features

Starters

- spring-boot-starter-security - Authentication and Authorization using Spring Security
- spring-boot-starter-data-jpa - Spring Data JPA with Hibernate
- spring-boot-starter-cache - Enabling Spring Framework's caching support
- spring-boot-starter-data-rest - Expose Simple REST Services using Spring Data REST

Inside Spring Boot Dependencies

```
<ehcache3.version>3.1.1</ehcache3.version>
...
<h2.version>1.4.192</h2.version>
<hamcrest.version>1.3</hamcrest.version>
<hazelcast.version>3.6.4</hazelcast.version>
<hibernate.version>5.0.9.Final</hibernate.version>
<hibernate-validator.version>5.2.4.Final</hibernate-validator.
<jackson.version>2.8.1</jackson.version>
....
<jersey.version>2.23.1</jersey.version>
<spring-security.version>4.1.1.RELEASE</spring-security.versio
<tomcat.version>8.5.4</tomcat.version>
<xml-apis.version>1.4.01</xml-apis.version>
```

Questions

- What is Starter Parent?
- What are the different things that are defined in Starter Parent?
- How does Spring Boot enforce common dependency management for all its Starter projects

Questions

- What is application.properties?
- What are some of the important things that can customized in application.properties?

309-Configuration with Application.Properties

application.properties | /Users/rangaraokaranam/.m2 | /Users/rangaraokaranam/.m2 | restful-web-services/pom.x | org.springframework.boot:sp | *applic

```
1 logging.level.org.springframework = debug
2 spring.jackson.serialization.write-dates-as-timestamps=false
3
4 management.security.enabled=false
5 security.basic.enabled=false
6
7 security.user.name=username
8 security.user.password=password
9
10 spring.jpa.show-sql=true
11 spring.h2.console.enabled=true
```

Example Configuration

```
logging.file=
logging.level.*=
spring.autoconfigure.exclude=
spring.profiles.active=
server.error.path=/error
server.port=8080
spring.http.converters.preferred-json-mapper=jackson
```

Questions

- What is a profile?
- How do you define beans for a specific profile?
- How do you create application configuration for a specific profile?

Profile

- application-dev.properties
- application-qa.properties
- application-stage.properties
- application-prod.properties
- application.properties

Setting a profile

- Using -Dspring.profiles.active=prod in VM Arguments
- In application.properties, spring.profiles.active=prod

Profile

- Based on the active profile, appropriate configuration is picked up.
- Used to Configure Resources - Databases, Queues, External Services

Profiles in code

@Profile("dev") on a bean

```
@Profile("dev")
@Bean
public String devBean() {
  return "I will be available in profile dev";
}

@Profile("prod")
@Bean
public String prodBean() {
  return "I will be available in profile prod";
}
```

Questions

- What is Spring Boot Actuator?
- How do you monitor web services using Spring Boot Actuator?
- How do you find more information about your application envrionment using Spring Boot?

Spring Boot Actuator

- Monitoring
  - /env, /metrics, /trace, /dump
  - /beans, / autoconfig, /configprops, /mappings
- URL has changed in Spring Boot 2.0 - /application

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```

Properties


```
{
  "status": "UP",
  "diskSpace": {
    "status": "UP",
    "total": 498887294976,
    "free": 233437093888,
    "threshold": 10485760
  },
  "db": {
    "status": "UP",
    "database": "H2",
    "hello": 1
  }
}
```


Links